

Online Dual Edge Coloring of Paths and Trees

Lene M. Favrholdt* and Jesper W. Mikkelsen*
 {lenem,jesperwm}@imada.sdu.dk

Department of Mathematics and Computer Science
 University of Southern Denmark

Abstract. We study a dual version of online edge coloring, where the goal is to color as many edges as possible using only a given number, k , of available colors. All of our results are with regard to competitive analysis. For paths, we consider $k = 2$, and for trees, we consider any $k \geq 2$. We prove that a natural greedy algorithm called **First-Fit** is optimal among deterministic algorithms on paths as well as trees. This is the first time that an optimal algorithm for online dual edge coloring has been identified for a class of graphs. For paths, we give a randomized algorithm, which is optimal and better than the best possible deterministic algorithm. Again, it is the first time that this has been done for a class of graphs. For trees, we also show that even randomized algorithms cannot be much better than **First-Fit**.

1 Introduction

In the classical edge coloring problem, the edges of a graph must be colored using as *few colors* as possible, under the constraint that no two adjacent edges receive the same color. There is a dual version of the problem where a fixed number, k , of colors is given and the goal is to color as *many edges* as possible, using at most k colors. Sometimes the classical problem is called the *minimization* version and the dual problem is called the *maximization* version of the problem.

In this paper, we study the online version of the maximization problem. In the online version, the edges of the graph arrive one by one, each specified by its endpoints. Immediately upon receiving an edge, the algorithm must either color the edge with one of the k colors or reject the edge. The decision of which of the k colors to use or to reject the edge is irrevocable. We call this problem **EDGE- k -COLORING**. For any class, **CLASS**, of graphs, we let **EDGE- k -COLORING(CLASS)** denote the problem of **EDGE- k -COLORING** restricted to graphs of class **CLASS**. For instance, **EDGE-2-COLORING(PATH)** is the online problem of coloring as many edges as possible in a path using only two colors.

Quality measure. We measure the quality of an online algorithm, **A**, for **EDGE- k -COLORING** using the standard notion of competitive ratio [10, 14]. The competitive ratio compares the performance of **A** to that of an optimal offline algorithm,

* Supported in part by the Villum Foundation and the Danish Council for Independent Research, Natural Sciences.

OPT. We denote by $A(\sigma)$ the number of edges colored by A when given a sequence, σ , of edges. Similarly, $OPT(\sigma)$ is the number of edges in σ colored by OPT. The algorithm A is said to be C -competitive if there exists a constant b such that $A(\sigma) \geq C \cdot OPT(\sigma) - b$ for any input sequence σ . The competitive ratio, $C_A(k)$, of A is the supremum over all C for which A is C -competitive. The competitive ratio of A for EDGE- k -COLORING(CLASS) is denoted by $C_A^{\text{CLASS}}(k)$.

Note that by this definition, $0 \leq C_A(k) \leq 1$. In particular, upper bounds on the competitive ratio are negative results and lower bounds are positive results.

If the inequality above holds even when $b = 0$, we say that A is *strictly C-competitive*. This gives rise to the notion of *strict competitive ratio*. The results in this paper are strongest possible in the sense that all positive results hold for the strict competitive ratio and all negative results hold for the competitive ratio.

For randomized algorithms, a similar definition of competitive ratio is used but $A(\sigma)$ is replaced by the expected value $E[A(\sigma)]$.

Notation and terminology. We label the k colors $1, 2, \dots, k$. For $1 \leq i \leq j \leq k$, define $C_{i,j} = \{i, i+1, \dots, j\}$. At any fixed point in the processing of the input sequence, we denote by C_v the set of colors used at edges incident to the vertex v . A color $i \in C_{1,k}$ is said to be *available* at v if $i \notin C_v$. Two colorings of a graph are said to be *equivalent* if one can be obtained from the other by renaming the colors.

If v is a vertex in the input graph, we denote by $d(v)$ the number of edges incident to v . An *isolated edge* $e = (v, u)$ is an edge such that $d(v) = d(u) = 1$ at the time where e is revealed. For any m , we let $\langle e_1, e_2, \dots, e_m \rangle$ denote a path with m edges and label the edges such that, for $2 \leq i \leq m-1$, e_i is adjacent to e_{i-1} and e_{i+1} . A *star* with m edges is the complete bipartite graph $K_{1,m}$.

Algorithms. An algorithm is called *fair* if it never rejects an edge unless all of the k colors have already been used on adjacent edges. In [7], the following two fair and deterministic algorithms were studied:

First-Fit (FF) uses the lowest available color when coloring an edge. It can be viewed as the natural greedy strategy.

Next-Fit (NF) remembers the last used color c_{last} . When coloring an edge, it uses the first available color in the ordered sequence $\langle c_{\text{last}} + 1, \dots, k, 1, \dots, c_{\text{last}} \rangle$. For the very first edge, it uses the color 1.

For the EDGE-2-COLORING(PATH) problem, we introduce a new family of randomized algorithms: For $\frac{1}{2} \leq p \leq 1$, **Rand_p** is defined as follows. Whenever an isolated edge is revealed, **Rand_p** uses the color 1 with probability p and the color 2 with probability $1 - p$. All non-isolated edges are colored (with the only remaining color) if possible. Note that **Rand₁** is identical to **First-Fit**.

Previous results. In [7] it is shown that any fair algorithm for EDGE- k -COLORING has a competitive ratio of at least $2\sqrt{3} - 3 \approx 0.464$, and at most $\frac{1}{2}$ if it is deterministic. The lower bound is tight in the sense that **Next-Fit** has a competitive ratio of exactly $2\sqrt{3} - 3$. It remains an open problem if any algorithm has a

competitive ratio better than $2\sqrt{3} - 3$. The authors of [7] also show that no algorithm (even when allowing randomization) has a competitive ratio better than $\frac{4}{7} \approx 0.57$.

The problem $\text{EDGE-}k\text{-COLORING}(k\text{-COLORABLE})$ is also studied in [7]. When the input graph is k -colorable, any fair algorithm is shown to have a competitive ratio of at least $\frac{1}{2}$. Again, the lower bound is tight because **Next-Fit** has a competitive ratio of $\frac{1}{2}$. The competitive ratio of **First-Fit** is shown to be $\frac{k}{2k-1}$. An upper bound of $\frac{2}{3}$ is given for deterministic algorithms in this case.

We remark that all of the negative results mentioned above hold even if the input graph is bipartite. Thus, contrary to offline edge coloring, the online $\text{EDGE-}k\text{-COLORING}$ problem does not appear to be significantly easier when restricted to bipartite graphs.

It is well known that for $k = 1$ (i.e., for the matching problem), the greedy algorithm is optimal with a competitive ratio of $\frac{1}{2}$.

The relative worst order ratio [3, 4] of both the maximization and minimization version of online edge coloring is studied in [6]. For the maximization version, it is shown that **First-Fit** and **Next-Fit** are not (strictly) comparable. This is true even when the input is restricted to bipartite graphs. For the minimization version, **First-Fit** is proven better than **Next-Fit**.

The minimization version of online edge coloring is studied in [1]. If an online algorithm never introduces a new color unless forced to do so, it will never use more than $2\Delta - 1$ different colors on graphs of maximum degree Δ . It is shown in [1] that no (randomized) online algorithm can do better than this, even if the input graph is restricted to being a forest. On any graph, an optimal offline algorithm uses at most $\Delta + 1$ colors, and on trees, Δ colors suffice. Hence, any algorithm that introduces a new color only when necessary, has a competitive ratio of 2, and this is optimal.

The problem of online vertex coloring has received much attention, especially in the minimization version (see [11] for a survey). Edge coloring a path of m edges is equivalent to vertex coloring a path of m vertices. Thus, our results for $\text{EDGE-2-COLORING}(\text{PATH})$ are also valid for online dual vertex coloring of paths with 2 colors available.

A study of approximation algorithms for the offline version of $\text{EDGE-}k\text{-COLORING}$ for multigraphs was initiated in [8]. This line of work has been continued in [5, 9, 12, 13] for both simple graphs and multigraphs.

Our contribution. For $\text{EDGE-2-COLORING}(\text{PATH})$, we give a $\frac{4}{5}$ -competitive randomized algorithm and prove that this is optimal. We also show that no deterministic algorithm can be better than $\frac{2}{3}$ -competitive and observe that this upper bound is tight, since **First-Fit** is $\frac{2}{3}$ -competitive. This is the first example of a class of graphs for which a randomized algorithm for $\text{EDGE-}k\text{-COLORING}$ is proven optimal and better than any deterministic algorithm.

For $\text{EDGE-}k\text{-COLORING}(\text{TREE})$ where $k \geq 2$, we prove that **First-Fit** is $\frac{k-1}{k}$ -competitive and that no deterministic or fair algorithm can be better than this. Thus, an algorithm would have to be both randomized and unfair to achieve

a better competitive ratio than **First-Fit**. However, we show that even such algorithms cannot be better than $\frac{k}{k+1}$ -competitive. We also show that any fair algorithm is $\frac{2\sqrt{k}-2}{2\sqrt{k}-1}$ -competitive and that the competitive ratio of **Next-Fit** is no better than this if k is a square number. This implies that the competitive ratio of any fair algorithm goes to 1 as k goes to infinity.

PATH and TREE are the first examples of graph classes for which an optimal deterministic algorithm for EDGE- k -COLORING has been identified.

2 A Charging Technique for Proving Positive Results

We will now describe a simple charging technique for proving lower bounds on the competitive ratio. The technique was first used for deterministic algorithms in [7]. For some C , $0 \leq C \leq 1$, our goal is to prove that a given (possibly randomized) algorithm **A** is C -competitive. Assume that the edges of a graph $G = (V, E)$ have been given in some order, σ , and let $E_{\text{OPT}} \subseteq E$ be the set of edges colored in some optimal solution.

The *initial value* $v_i(e)$ of an edge, $e \in E$, is $v_i(e) = \Pr[e \text{ is colored by } \mathbf{A}]$. For deterministic algorithms, $v_i(e) \in \{0, 1\}$ for all $e \in E$. Note that by linearity of expectation, we have $E[\mathbf{A}(\sigma)] = \sum_{e \in E} v_i(e)$.

The *surplus* $v_+(e)$ of an edge, $e \in E$, (with respect to C) is

$$v_+(e) = \begin{cases} v_i(e) - C, & \text{if } e \in E_{\text{OPT}} \\ v_i(e), & \text{if } e \notin E_{\text{OPT}} \end{cases}$$

We let $E_+ \subseteq E$ and $E_- \subseteq E$ denote the sets of edges with positive and negative surplus, respectively. Clearly, $E_- \subseteq E_{\text{OPT}}$. For deterministic algorithms, E_- is exactly those edges in E_{OPT} that are not colored by the algorithm, and E_+ is the set of edges colored by the algorithm (assuming $C < 1$). The total positive surplus $\sum_{e \in E_+} v_+(e)$ will be redistributed among the edges in E_- according to some strategy. This strategy is what needs to be defined when applying the technique.

The *final value* $v_f(e)$ of an edge $e \in E_{\text{OPT}}$ is the total value of e after the redistribution of surplus. Since only surplus value is redistributed, $v_f(e) \geq C$ for all $e \in E_{\text{OPT}} \setminus E_-$. Thus, if it can be proven that $v_f(e) \geq C$ for all $e \in E_-$, then

$$E[\mathbf{A}(\sigma)] = \sum_{e \in E} v_i(e) = \sum_{e \in E} v_f(e) \geq \sum_{e \in E_{\text{OPT}}} v_f(e) \geq C \cdot \text{OPT}(\sigma).$$

Thus, it follows that **A** is (strictly) C -competitive.

3 Coloring of Paths

In this section, we study the EDGE- k -COLORING problem when the input graph is a path. Clearly, this is only interesting if $k \leq 2$. In this paper, we consider

solely the case where $k = 2$, but we remark that one can use the same techniques to obtain tight bounds on the competitive ratio when $k = 1$. Also, the results for PATH can be extended to graphs of maximum degree 2.

For EDGE-2-COLORING(PATH), our main result is a randomized algorithm with a competitive ratio of $\frac{4}{5}$ and a proof that this is optimal. Before considering randomized algorithms, we give tight lower and upper bounds on the competitive ratio of deterministic algorithms. For 2-colorable graphs, the ratios of Propositions 1 and 2 both follow from [7]. Clearly, the positive results carry over to paths, but for $k = 2$, the graphs used in [7] for the negative results are not connected. In Propositions 1 and 2, we give simple proofs that the negative results are also valid when the graph is a path.

Proposition 1. *For EDGE-2-COLORING(PATH), **Next-Fit** is a worst possible fair algorithm with*

$$C_{NF}^{\text{PATH}}(2) = \frac{1}{2}.$$

Proof. The lower bound for fair algorithms follows, since each rejected edge is adjacent to exactly two colored edges, and each colored edge is adjacent to at most two rejected edges.

For the upper bound, consider a path $\langle e_1, \dots, e_{2m+1} \rangle$ with $2m + 1$ edges. The adversary first reveals the odd-numbered edges in order of increasing indices. **Next-Fit** will alternate between the two colors. Afterwards, the adversary reveals all the even-numbered edges. These edges must all be rejected by **Next-Fit**. Thus, the competitive ratio of **Next-Fit** is at most $\frac{m+1}{2m+1}$ which tends to $\frac{1}{2}$ as m tends to infinity. \square

Proposition 2. *For EDGE-2-COLORING(PATH), **First-Fit** is an optimal deterministic algorithm with*

$$C_{FF}^{\text{PATH}}(2) = \frac{2}{3}.$$

Proof. Since a path is 2-colorable, the lower bound for **First-Fit** follows from a result in [7] stating that the competitive ratio of **First-Fit** is $\frac{k}{2k-1}$ for the EDGE- k -COLORING(k -COLORABLE) problem.

For the upper bound, let D be a deterministic algorithm and let $n \in \mathbb{N}$. The adversary first gives n disjoint paths of length two. Call these the *initial paths*. Let $F = \{f_1, \dots, f_{n_1}\}$ be the set of those initial paths in which both edges have been colored by D and let $U = \{u_1, \dots, u_{n_2}\}$ be the set of those initial paths in which at least one edge has been rejected.

In each path in F , both colors 1 and 2 are represented. The adversary reveals an edge connecting the edge with the color 1 in the path f_i to the edge with the color 2 in the path f_{i+1} , for $1 \leq i < n_1$. These connecting edges must be rejected by D so the number of colored edges in this component is at most $2n_1$. The adversary also reveals an edge connecting u_i to u_{i+1} , for $1 \leq i < n_2$. Even if all of these connecting edges can be colored, the number of colored edges in this component is at most $2n_2 - 1$.

Finally, if both F and U are non-empty, the adversary connects the two constructed paths by a single edge which may possibly be colored. It follows that the number of colored edges can be at most $2n_1 + (2n_2 - 1) + 1 = 2n$. Since the total number of edges is $3n - 1$, we get an upper bound on the competitive ratio of $\frac{2n}{3n-1}$ which tends to $\frac{2}{3}$ as n tends to infinity. \square

Knowing that no deterministic algorithm can be better than $\frac{2}{3}$ -competitive, a natural question to ask is how good a randomized algorithm can be. To this end, we analyze the family of fair, randomized algorithms, \mathbf{Rand}_p , defined in the introduction.

Theorem 1. *Let $\frac{1}{2} \leq p \leq 1$. Then,*

$$C_{\mathbf{Rand}_p}^{\text{PATH}}(2) = \min \left\{ p^2 - p + 1, \frac{2}{3}(-p^2 + p + 1) \right\}.$$

Proof. We first show the upper bound. The adversary will reveal the edges of a path $P = \langle e_1, \dots, e_m \rangle$ with m edges. Consider the following two adversary strategies for doing so:

- (i) The adversary first reveals all edges e_i with $i \equiv 1 \pmod{3}$, followed by all edges e_i with $i \equiv 0 \pmod{3}$. Finally, all the remaining edges are revealed.
- (ii) The adversary first reveals all the odd numbered edges and thereafter all the even numbered edges.

If the adversary uses strategy (i), it chooses m such that 3 divides $m - 1$. Note that each edge e_i with $i \equiv 2 \pmod{3}$ has probability $p(1 - p) + (1 - p)p$ of being colored. It follows that

$$\begin{aligned} E[\mathbf{Rand}_p(P)] &= \frac{2}{3}(m - 1) + \frac{2}{3}(m - 1)(1 - p)p + 1 \\ &= \frac{2}{3}(-p^2 + p + 1)(m - 1) + 1. \end{aligned}$$

If the adversary uses strategy (ii), it makes sure that the number, m , of edges in P is odd. Note that each even numbered edge has probability $p^2 + (1 - p)^2$ of being colored. It follows that

$$\begin{aligned} E[\mathbf{Rand}_p(P)] &= \frac{1}{2}(m - 1) + \frac{1}{2}(m - 1)(p^2 + (1 - p)^2) + 1 \\ &= (p^2 - p + 1)(m - 1) + 1. \end{aligned}$$

Thus, if $\frac{2}{3}(-p^2 + p + 1) \leq p^2 - p + 1$, the adversary uses strategy (i) and otherwise it uses strategy (ii). By choosing m sufficiently large, this proves the upper bound.

For the lower bound, fix $\frac{1}{2} \leq p \leq 1$. Let P be a path and assume that the edges of P are given to \mathbf{Rand}_p in some order. Consider an edge e at the time of its arrival. If two edges adjacent to e have already been revealed, we say that e

is a *critical edge*. Denote by E_{crit} the critical edges of P . Note that since Rand_p is fair, it will never reject an edge which is not critical.

We will apply the charging technique described in Section 2. That is, we will define a strategy for distributing the total surplus among the edges of the path and determine the largest possible C such that all edges receive a final value of at least C . This will imply that Rand_p is C -competitive. Note that all non-critical edges have an initial value of 1 and, hence, a surplus of $1 - C$. Thus, $E_- \subseteq E_{\text{crit}}$.

Let e be a non-critical edge. Consider the largest connected component P_e induced by edges from $E \setminus E_{\text{crit}}$ containing e . Let e_{first} be the edge in P_e which was revealed first. We define $l(e)$ to be the length of the shortest path in P_e containing e and e_{first} . If e is revealed as an isolated edge, then $l(e) = 1$. We say that e is *odd* if $l(e)$ is odd and that e is *even* if $l(e)$ is even.

Fact: If $l(e)$ is odd, then the probability of e being colored with the color 1 is p . If $l(e)$ is even, then the probability of e being colored with the color 1 is $1 - p$. This is easily proven by induction on $l(e)$.

Let e_{crit} be a critical edge. Denote by e_l and e_r the two edges adjacent to e_{crit} . These must both be non-critical and thus must be colored by Rand_p . The edge e_{crit} will be rejected if and only if e_l and e_r are colored with different colors. Note that the random variable denoting the color received by e_l is independent of the random variable denoting the color received by e_r . We now consider two cases:

Case 1: One of e_l and e_r is odd and the other is even. Without loss of generality, assume that e_l is odd and that e_r is even. By the fact stated above, the probability of e_{crit} being colored is $p(1 - p) + (1 - p)p$. Since e_r is even, it must be adjacent to at least one non-critical edge e'_r . We transfer a value of $\frac{1}{2}(1 - C)$ from each of e_l and e'_r to e_{crit} and a value of $1 - C$ from e_r to e_{crit} . Transferring the entire surplus of $1 - C$ from e_r to e_{crit} is possible, since e'_r is non-critical and therefore e_{crit} is the only critical edge adjacent to e_r . Thus, the final value of e_{crit} is $2p(1 - p) + 2(1 - C)$. Since

$$2p(1 - p) + 2(1 - C) \geq C \Leftrightarrow \frac{2}{3}(-p^2 + p + 1) \geq C, \quad (1)$$

it follows that if C is at most $\frac{2}{3}(-p^2 + p + 1)$, then the final value of e_{crit} is at least C .

Case 2: e_l and e_r are both odd or both even. In this case, the probability of e_{crit} being colored is $p^2 + (1 - p)^2$. It follows that $v_i(e_{\text{crit}}) = p^2 + (1 - p)^2$. Since e_l and e_r are non-critical, they both have a surplus of $1 - C$. We will transfer a value of $\frac{1}{2}(1 - C)$ from each of them to the critical edge e_{crit} . Thus, the final value of e_{crit} is $p^2 + (1 - p)^2 + (1 - C) = 2(p^2 - p + 1) - C$. Since

$$2(p^2 - p + 1) - C \geq C \Leftrightarrow p^2 - p + 1 \geq C, \quad (2)$$

it follows that if C is at most $p^2 - p + 1$, then the final value of e_{crit} is at least C .

We conclude that if C is bounded from above by both $p^2 - p + 1$ and $\frac{2}{3}(-p^2 + p + 1)$ then, using the strategy described above, all edges in the path receive a final value of at least C . \square

Theorem 1 shows that, for $p = \varphi/\sqrt{5}$, Rand_p has a competitive ratio of $\frac{4}{5}$ (where $\varphi = (1 + \sqrt{5})/2$ is the golden ratio). We will now show that $\frac{4}{5}$ is the best possible competitive ratio of *any* algorithm.

Theorem 2. *If R is an algorithm for EDGE-2-COLORING(PATH), then*

$$C_R^{\text{PATH}}(2) \leq \frac{4}{5}.$$

Proof. We will use Yao's minimax principle [2, 15]. Let $a = 3^b$, for some $b \in \mathbb{N}$, and consider a path $P = \langle e_1, e_2, \dots, e_{a-2} \rangle$. We will describe a probability distribution \mathcal{P} over all permutations of the edges of P . This will be done by describing a randomized adversary.

The edges are given in three phases. In the first two phases, isolated edges are given.

- The first (possibly empty) phase is divided into L subphases, $0 \leq L \leq b-1$, where L is a random variable with $\Pr_{\mathcal{P}}[L = i] = (\frac{1}{2})^{i+1}$, for $0 \leq i \leq b-2$, and $\Pr_{\mathcal{P}}[L = b-1] = (\frac{1}{2})^{b-1}$. Thus, the edges of Subphase 1 are given with probability $\frac{1}{2}$, and after giving the edges of Subphase i , the adversary adds another subphase with probability $\frac{1}{2}$, if $i < b-1$, and with probability 0, if $i = b-1$. Note that $\Pr_{\mathcal{P}}[L > i] = \Pr_{\mathcal{P}}[L = i] = (\frac{1}{2})^{i+1}$, for $0 \leq i \leq b-2$. In Subphase i , $a_i = a/3^i$ isolated edges are given. We let $N_i = \sum_{j=1}^i a_j$ denote the number of edges given in the first i subphases. In particular, $N_0 = 0$. The set of edges given in the Subphase i is denoted by $E_i = \{e_1^i, e_2^i, \dots, e_{a_i}^i\}$, where $e_j^i = e_{2N_{i-1}+2j-1}$. Thus, in this phase, every second of the first $2N_L$ edges in the path is given.
- In the second phase, every third edge of the remaining part of the path is given. These are the edges $E_{L+1} = \{e_1^{L+1}, e_2^{L+1}, \dots, e_{a_{L+1}}^{L+1}\}$, where $a_{L+1} = a/3^{L+1}$ and $e_j^{L+1} = e_{2N_L+3j-2}$.
- In the third phase, all the remaining edges of P are given. For $1 \leq i \leq L$, we let F_i denote those edges that connect two edges of E_i . Moreover, we let F_{L+1} denote the set of edges incident to an edge of E_{L+1} and no edge of E_L .

Note that the total number of edges given is as stated above:

$$\begin{aligned} |P| &= \sum_{j=1}^L (|E_j| + |F_j| + 1) + |E_{L+1}| + |F_{L+1}| \\ &= 2 \sum_{j=1}^L \frac{a}{3^j} + 3 \frac{a}{3^{L+1}} - 2 = \left(1 - \frac{1}{3^L}\right) a + \frac{a}{3^L} - 2 = a - 2. \end{aligned}$$

This finishes the description of the probability distribution \mathcal{P} .

Let \mathbf{D} be any deterministic algorithm. We will show that $E_{\mathcal{P}}[\mathbf{D}(P)]$ is at most $\frac{4}{5}a + O(1)$. By Yao's principle, this will complete the proof.

We first introduce some further terminology. For any $1 \leq i \leq L+1$ and $2 \leq j \leq a_i$, we define $\text{prev}(e_j^i) = e_{j-1}^i$. We say that $\text{prev}(e_j^i)$ is the *previous* isolated

edge of e_j^i . Note that no edge is the previous isolated edge of e_1^i , $1 \leq i \leq L+1$. Let E_i^s be those edges in E_i which are colored with the same color as the previous isolated edge. Define E_i^d similarly, but for edges colored with a different color than the previous isolated edge. Let E_i^r be those edges of E_i which are rejected. Finally, let E_i^c be those edges of E_i which are colored but where the previous isolated edge is rejected. Clearly, $|E_i^c| \leq |E_i^r|$.

For $1 \leq i \leq L$, let X_i^r be a random variable denoting the number of rejected edges in $E_i \cup F_i$. We will give a lower bound on $E_{\mathcal{P}}[X_i^r]$. For each isolated edge e_j^i with $1 \leq i \leq L+1$ and $2 \leq j \leq a$, consider the probability of at least one of e_j^i and the edge(s) connecting e_j^i to $\text{prev}(e_j^i)$ being rejected. For each edge in E_i^s , the algorithm \mathcal{D} makes a rejection with probability $\frac{1}{2}$, since it will be forced to do so if $i = L+1$. Conversely, for each edge in E_i^d the algorithm \mathcal{D} makes a rejection with probability $\frac{1}{2}$, since it is forced to do so if $i \leq L$. Also, for each edge in E_i^r the algorithm \mathcal{D} makes a rejection with probability 1. Recall that $\Pr_{\mathcal{P}}[i \leq L+1] = \Pr_{\mathcal{P}}[L > i-2] = \frac{1}{2^{i-1}}$. Hence,

$$\begin{aligned} E_{\mathcal{P}}[X_i^r] &\geq \frac{1}{2^{i-1}} \left(\frac{1}{2} |E_i^s| + \frac{1}{2} |E_i^d| + |E_i^r| \right) \geq \frac{1}{2^i} (|E_i^s| + |E_i^d| + |E_i^r| + |E_i^c|) \\ &= \frac{1}{2^i} (|E_i| - 1) = \frac{1}{2^i} \left(\frac{a}{3^i} - 1 \right) = \frac{a}{6^i} - \frac{1}{2^i}, \end{aligned}$$

where the second inequality follows from $|E_i^r| \geq |E_i^c|$. Thus,

$$\sum_{i=1}^b E_{\mathcal{P}}[X_i^r] \geq \sum_{i=1}^b \left(\frac{1}{6^i} a - \frac{1}{2^i} \right) > \frac{1}{5} a - \frac{1}{5a^{\log_3(6)-1}} - 1$$

Finally, since \mathcal{OPT} can color all $a-2$ edges of the path, we get that

$$E_{\mathcal{P}}[\mathcal{D}(P)] \leq \frac{4}{5} a + \frac{1}{5a^{\log_3(6)-1}} + 1 = \frac{4}{5} a + O(1) = \frac{4}{5} \mathcal{OPT}(P) + O(1).$$

□

Theorems 1 and 2 together give the following corollary.

Corollary 1. For $p = \frac{\varphi}{\sqrt{5}}$, Rand_p is optimal for $\text{EDGE-2-COLORING}(\text{PATH})$ with

$$C_{\text{Rand}_p}^{\text{PATH}}(2) = \frac{4}{5}.$$

4 Coloring of Trees

We will now consider the $\text{EDGE-}k\text{-COLORING}$ problem when the input graph is a tree. Our main result is a proof that **First-Fit** is an optimal deterministic algorithm. We also show that, for any fixed $k \geq 4$, **First-Fit** has a better competitive ratio than **Next-Fit** for $\text{EDGE-}k\text{-COLORING}(\text{TREE})$. First, we give a general upper bound for algorithms that are deterministic or fair.

Theorem 3. *If A is a deterministic or fair algorithm and $k \geq 2$, then*

$$C_A^{\text{TREE}}(k) \leq \frac{k-1}{k}.$$

Proof. The adversary reveals the edges of a tree in N steps, for some large $N \in \mathbb{N}$. The set of edges revealed in the i th step constitute a star, S_i , with $k+1$ edges and center vertex c_i . If at least one edge in S_{i-1} is colored, the adversary chooses $c_i = x$ for some colored edge (c_{i-1}, x) in S_{i-1} . Otherwise, it chooses $c_i = x$ for an arbitrary edge (c_{i-1}, x) in S_{i-1} . Note that the adversary is clearly able to identify a colored edge in S_{i-1} , if one exists: If A is deterministic, this is trivially true, and if A is fair, the first $k-1$ edges of S_{i-1} will be colored.

The algorithm A may color k edges of S_1 . For all other values of i , there are two possibilities:

- If A colors even a single edge of S_{i-1} , then it can color at most $k-1$ edges of S_i .
- Even if A rejects all edges of S_{i-1} , then it can color at most k edges of S_i .

Let N_0 denote the number of stars where A colors no edges. Then, A colors at most $(N_0+1)k + (N-2N_0-1)(k-1) = N(k-1) - (k-2)N_0 + 1 \leq N(k-1) + 1$ edges. On the other hand, in each star, OPT colors the k edges not incident to other stars, in total Nk edges. Since N can be arbitrarily large, this shows that the competitive ratio of A is at most $\frac{k-1}{k}$. \square

Using the charging technique of Section 2, we will show that Theorem 3 is tight by proving a matching lower bound for **First-Fit**. To this end, we introduce some terminology related to deterministic algorithms.

Let A be a deterministic algorithm for **EDGE- k -COLORING**, let $G = (V, E)$ be a graph, and suppose that A has been given the edges of G in some order. Recall that, since A is deterministic, E_+ denotes the set of edges colored by A , and E_- denotes the set of edges colored by OPT only. We partition E_+ into the set, E_+^d , of edges colored by both A and OPT (*double colored* edges) and the set, E_+^s , of edges colored by A only (*single colored* edges). Thus, $E_{\text{OPT}} = E_- \cup E_+^d$. For $x \in V$, let $E_+(x)$ be the edges in E_+ incident to x and let $d_+(x) = |E_+(x)|$. Define $E_-(x)$, $E_+^d(x)$, $E_+^s(x)$, $d_-(x)$, $d_+^d(x)$ and $d_+^s(x)$ similarly.

Theorem 4. *For $k \geq 2$, **First-Fit** is an optimal deterministic algorithm for **EDGE- k -COLORING(TREE)** with*

$$C_{\text{FF}}^{\text{TREE}}(k) = \frac{k-1}{k}.$$

Proof. Fix a tree $T = (V, E)$ and assume that the edges of E have been revealed to **First-Fit** in some order. For the analysis, we will view T as a rooted tree by choosing an arbitrary vertex to be the root. When writing $e = (x, y) \in E$, we imply that x is the parent vertex of y .

Following Section 2, we set $C = \frac{k-1}{k}$. An edge in E_+^d then has a surplus of $1 - C = \frac{1}{k}$ and an edge in E_+^s has a surplus of 1. On the other hand, an edge in E_- has an initial value of zero.

We will define a strategy to distribute the total positive surplus obtained by **First-Fit** among the edges in E_- such that each edge gets a final value of at least C . For ease of presentation, the strategy will be described in a stepwise manner:

- Step 1: Consider in turn all edges $e = (v, u) \in E_+$. Let c be the color assigned to e by **First-Fit** and let $e' = (w, v)$ be the parent edge of e (if it exists).
- (a) If $e' \in E_+^d$ and e' has been colored with a color $c' > c$, then e transfers a value of $\frac{1}{k}$ to w .
 - (b) Any surplus remaining at e is transferred to v .
- For each vertex v , let $m(v)$ denote the value transferred to v in this step.
- Step 2: Consider in turn all vertices $v \in V$.
- (a) If v has a parent edge $e' \in E_-$, then v transfers a value of $\min \{m(v), \frac{k-1}{k}\}$ to e' .
 - (b) Any value remaining at v is distributed equally among the child edges of v belonging to E_- .

The following simple but useful properties of the strategy defined above will be used to prove the theorem. Each of the four facts gives a lower bound on the value transferred from an edge $e = (v, u)$ to its parent vertex, v . Let $e' = (w, v)$ be the parent edge of e (if it exists).

Fact 1: Assume that $e \in E_+^s$. If $e' \notin E_+^d$ or e' does not exist, then e contributes a value of 1 to $m(v)$. If $e' \in E_+^d$, then e contributes a value of at least $\frac{k-1}{k}$ to $m(v)$. If $e' \notin E_+^d$ or e' does not exist, then e transfers a value of 1 to v according in Step 1(b). Conversely, If $e' \in E_+^d$, then e transfers a value of at most $\frac{1}{k}$ to w in Step 1(a) and hence e transfers a value of at least $\frac{k-1}{k}$ to v in Step 1(b).

*Fact 2: Assume that e is colored with the color c . If $e' \notin E_+^d$ or e' does not exist, then $m(v) \geq \frac{c}{k}$. If $e \in E_+^s$, this follows immediately from Fact 1. Otherwise, note that by the definition of **First-Fit**, it must hold that $\mathcal{C}_{1,c} \subseteq \mathcal{C}_v \cup \mathcal{C}_u$. In Step 1, the edges incident to v and u colored with a color in $\mathcal{C}_{1,c}$ each transfers a value of at least $\frac{1}{k}$ to v .*

Fact 3: Assume that $e \in E_+^d$. If $e' \notin E_+^d$, then e contributes a value of $\frac{1}{k}$ to $m(v)$. This follows, since e does not transfer any value to w in Step 1(a).

In order to state the next fact, we need to introduce some new terminology. For $v \in V$, let $\widehat{c}_v = \max \{\overline{\mathcal{C}}_v \cup \{0\}\}$. That is, \widehat{c}_v is the largest color available at v (and $\widehat{c}_v = 0$ if no colors are available). If an edge e incident to v is colored with a color $c > \widehat{c}_v$, then e is said to be a *high-colored* edge (with respect to v). There must be exactly $k - \widehat{c}_v$ high-colored edges incident to v .

*Fact 4: Assume that $e \in E_+^d$. If e is high-colored with respect to v , then the colored child edges of e contribute a total value of at least $\frac{k-d_+(v)}{k}$ to $m(v)$. We will now prove Fact 4. Since e is high-colored, it follows from the definition of **First-Fit** that all colors in $\overline{\mathcal{C}}_v$ are represented at child edges of u . Thus, e has at least $|\overline{\mathcal{C}}_v| = k - d_+(v)$ child edges with lower colors than the color of e . Since $e \in E_+^d$, each of these child edges transfers a value of $\frac{1}{k}$ to v in Step 1(b).*

We will combine these facts to show that any edge $e = (x, y) \in E_-$ gets a final value of at least $\frac{k-1}{k}$. If $\mathcal{C}_x = \mathcal{C}_{1,k}$, then $\widehat{c}_x = 0$. Otherwise, $\widehat{c}_x \in \mathcal{C}_y$, since

First-Fit is fair. Hence, Fact 2 implies that $m(y) \geq \frac{\widehat{c}_x}{k}$. Thus, e receives a value of at least $\min\{\frac{k-1}{k}, \frac{\widehat{c}_x}{k}\}$ from y . In particular, we will assume that $\widehat{c}_x < k-1$, since otherwise we are done. We will now turn to proving that e receives a value of at least $\frac{k-\widehat{c}_x-1}{k}$ from x . This will finish the proof, since it means that e gets a final value of at least $\frac{\widehat{c}_x}{k} + \frac{k-\widehat{c}_x-1}{k} = \frac{k-1}{k}$.

Let $e' = (z, x)$ be the parent edge of x (if it exists). The rest of the proof is split into three cases depending on which of the sets E_d , E_- and E_+^s (if any) that contains e' .

Case 1: $e' \in E_+^d$. Recall that there are $k - \widehat{c}_x$ high-colored edges incident to x . Thus, x has at least $k - \widehat{c}_x - 1$ high-colored child edges, and at least $k - \widehat{c}_x - 1 - d_+^s(x)$ of them belong to E_+^d . By Fact 4, x receives a value of at least $\frac{k-d_+(x)}{k}$ from the child edges of each of these at least $k - \widehat{c}_x - 1 - d_+^s(x)$ edges. Moreover, by Fact 1, each of the $d_+^s(x)$ child edges of e' belonging to E_+^s contributes a value of $\frac{k-1}{k}$ to $m(x)$. Thus,

$$\begin{aligned}
m(x) &\geq (k - \widehat{c}_x - 1 - d_+^s(x)) \frac{k - d_+(x)}{k} + d_+^s(x) \frac{k-1}{k} \\
&= (k - \widehat{c}_x - 1 - d_+^s(x)) \frac{k - d_+(x)}{k} + d_+^s(x) \left(\frac{k - d_+(x)}{k} + \frac{d_+(x) - 1}{k} \right) \\
&= (k - \widehat{c}_x - 1) \frac{k - d_+(x)}{k} + d_+^s(x) \frac{d_+(x) - 1}{k} \\
&\geq (k - \widehat{c}_x - 1) \frac{k - d_+(x)}{k} + d_+^s(x) \frac{k - \widehat{c}_x - 1}{k} \\
&= (k - d_+(x) + d_+^s(x)) \frac{k - \widehat{c}_x - 1}{k} \\
&= (k - d_+^d(x)) \frac{k - \widehat{c}_x - 1}{k} \\
&\geq d_-(x) \frac{k - \widehat{c}_x - 1}{k}
\end{aligned}$$

Hence, since no value is transferred from x to e' in Step 2(a), each child edge of x belonging to E_- receives a value of at least $\frac{k-\widehat{c}_x-1}{k}$ from x in Step 2(b).

Case 2: $e' \in E_+^s$ or e' does not exist. In this case, since $e' \notin E_+^d$, x has at least $k - \widehat{c}_x - d_+^s(x)$ high-colored child edges belonging to E_+^d . By Fact 4, x receives a value of at least $\frac{k-d_+(x)}{k}$ from the child edges of each of these edges. Note that this value comes solely from child edges of x 's high-colored child edges, not from the high-colored edges themselves. Therefore, by Fact 3, there is also a contribution of $\frac{1}{k}$ from each of x 's child edges belonging to E_+^d . Finally, there are at least $d_+^s(x) - 1$ child edges of x belonging to E_+^s (if e' exists, there are $d_+^s(x) - 1$ such edges, and otherwise there $d_+^s(x)$ such edges). By Fact 1, each of

these edges transfers a value of 1 to x . Thus,

$$\begin{aligned}
m(x) &\geq (k - \widehat{c}_x - d_+^s(x)) \frac{k - d_+(x)}{k} + (d_+^s(x) - 1) + \frac{d_+^d(x)}{k} \\
&= (k - \widehat{c}_x - d_+^s(x)) \frac{k - d_+(x)}{k} + (d_+^s(x) - 1) \left(\frac{k - d_+(x)}{k} + \frac{d_+(x)}{k} \right) + \frac{d_+^d(x)}{k} \\
&= (k - \widehat{c}_x - 1) \frac{k - d_+(x)}{k} + (d_+^s(x) - 1) \frac{d_+(x)}{k} + \frac{d_+^d(x)}{k} \\
&= (k - \widehat{c}_x - 1) \frac{k - d_+(x)}{k} + (d_+^s(x) - 1) \frac{d_+(x) - 1}{k} + \frac{d_+^s(x) - 1 + d_+^d(x)}{k} \\
&= (k - \widehat{c}_x - 1) \frac{k - d_+(x)}{k} + (d_+^s(x) - 1) \frac{d_+(x) - 1}{k} + \frac{d_+(x) - 1}{k} \\
&= (k - \widehat{c}_x - 1) \frac{k - d_+(x)}{k} + d_+^s(x) \frac{d_+(x) - 1}{k} \\
&\geq d_-(x) \frac{k - \widehat{c}_x - 1}{k}, \text{ as in Case 1}
\end{aligned}$$

Hence, since no value is transferred from x to e' in Step 2(a), each child edge of x belonging to E_- receives a value of at least $\frac{k - \widehat{c}_x - 1}{k}$ from x in Step 2(b).

Case 3: $e' \in E_-$. The only difference to Case 2 is that x has exactly $d_+^s(x)$ child edges belonging to E_+^s . Thus,

$$\begin{aligned}
m(x) &\geq (k - \widehat{c}_x - d_+^s(x)) \frac{k - d_+(x)}{k} + d_+^s(x) + \frac{d_+^d(x)}{k} \\
&\geq d_-(x) \frac{k - \widehat{c}_x - 1}{k} + 1, \text{ using the same calculations as in Case 2}
\end{aligned}$$

Hence, since the value transferred from x to e' is smaller than 1, each child edge of x belonging to E_- receives a value larger than $\frac{k - \widehat{c}_x - 1}{k}$ from x . \square

By Theorems 3 and 4, an algorithm for EDGE- k -COLORING(TREE) can only be better than **First-Fit**, if it is both randomized and unfair. However, the next result shows that even such algorithms cannot do much better than **First-Fit**.

Theorem 5. *If R is an algorithm for EDGE- k -COLORING and $k \geq 2$, then*

$$C_R^{\text{TREE}}(k) \leq \frac{k}{k+1}.$$

Proof. The adversary first reveals the edges of a path $P = \langle e_1, \dots, e_m \rangle$, for some large $m \in \mathbb{N}$. Let v_1, \dots, v_{m+1} be the vertices in the path such that $e_i = (v_i, v_{i+1})$, for $1 \leq i \leq m$. If $E[R(P)] \leq \frac{k}{k+1}m$, the adversary reveals no more edges. If $E[R(P)] > \frac{k}{k+1}m$, then for each i , $1 \leq i \leq m+1$, the adversary reveals k edges constituting a star, S_i , with center vertex v_i . Let S be the set consisting of the edges of every star S_i for $1 \leq i \leq m+1$.

If the adversary only reveals the edges of the path P , then $E[R(P)] \leq \frac{k}{k+1}m$ and so $E[R(P)] \leq \frac{k}{k+1} \text{OPT}(P)$. Indeed, OPT can color all m edges in P , since

$k \geq 2$ and so $\text{OPT}(P) = m$. Assume now that the adversary also reveals the stars. In this case, OPT rejects all edges of the path and instead colors the k edges of every star. Thus, $\text{OPT}(P \cup S) = k(m+1)$. Note that each of the edges $e_i = (v_i, v_{i+1})$ is incident to the center vertices of both S_i and S_{i+1} . This implies that $E[\mathbf{R}(S)] \leq k(m+1) - 2E[\mathbf{R}(P)]$. Using the assumption $E[\mathbf{R}(P)] > \frac{k}{k+1}m$, we get that

$$\begin{aligned}
E[\mathbf{R}(P \cup S)] &= E[\mathbf{R}(P)] + E[\mathbf{R}(S)] \\
&\leq E[\mathbf{R}(P)] + k(m+1) - 2E[\mathbf{R}(P)] \\
&\leq k(m+1) - \frac{k}{k+1}m \\
&= \frac{k(km+k+1)}{k+1} \\
&= \frac{k}{k+1}k(m+1) + \frac{k}{k+1} \\
&= \frac{k}{k+1}\text{OPT}(P \cup S) + \frac{k}{k+1}
\end{aligned}$$

This shows that \mathbf{R} cannot be better than $\frac{k}{k+1}$ -competitive. \square

We now show that, for any fixed $k \geq 4$, **First-Fit** is better than **Next-Fit**, but the competitive ratio of any fair algorithm tends to 1 as k tends to infinity.

Theorem 6. *If F is a fair algorithm, then for any $k \geq 2$,*

$$C_F^{\text{TREE}}(k) \geq \frac{2\sqrt{k}-2}{2\sqrt{k}-1}.$$

Proof. Assume first that \mathbf{F} is a deterministic algorithm. Let $T = (E, V)$ be a tree and assume that the edges of T have been revealed to \mathbf{F} in some order. For the analysis, we will view T as a rooted tree by choosing an arbitrary vertex to be the root. As in the proof of Theorem 4, we let $e = (x, y)$ imply that x is the parent of y .

We will apply the charging technique from Section 2 to show that \mathbf{F} is C -competitive, where $C = \frac{2\sqrt{k}-2}{2\sqrt{k}-1}$. We will use the notation introduced on page 10 just before Theorem 4. Recall that all edges in E_+ have an initial value of 1. Edges in E_+^d have a surplus of $1 - C$ and edges in E_+^s have a surplus of 1. Edges in E_- have an initial value of 0. The goal is to distribute the surplus from E_+^d and E_+^s among the edges in E_- so that all of them get a final value of at least C . To this end, we use the following strategy:

Step 1: Each edge $(v, u) \in E_+$ transfers its entire surplus to its parent vertex, v .

For each vertex v , let $m(v)$ denote the value transferred to v in this step.

Step 2: Consider in turn all vertices $v \in V$.

(a) If v has a parent edge $e' \in E_-$, then v transfers a value of $\min\{m(v), C\}$ to e' .

- (b) Any value remaining at v is distributed equally among the child edges of v belonging to E_- .

For each edge e , let $m_v(e)$ denote the value transferred from v to e in this step.

This finishes the description of the strategy.

Fix an edge $e = (x, y) \in E_-$. According to Step 1 and Step 2(a), we must have $m_y(e) = \min\{C, d_+(y) - Cd_+^d(y)\}$. We need to show that $m_x(e) + m_y(e) \geq C$. According to the strategy, this is always the case unless $d_+(y) - Cd_+^d(y) < C$. Note that

$$\begin{aligned} d_+(y) - Cd_+^d(y) < C &\Rightarrow d_+(y) < C(d_+^d(y) + 1) < (d_+^d(y) + 1) \\ &\Rightarrow d_+(y) - d_+^d(y) < 1 \\ &\Rightarrow d_+(y) = d_+^d(y). \end{aligned}$$

It follows that we only need to consider the case where $d_+(y) = d_+^d(y)$, meaning that all of the edges incident to y which have been colored by **F** have also been colored by **OPT**. This implies that the value transferred to e from its colored child edges is

$$m_y(e) = (1 - C)d_+(y).$$

When calculating a lower bound on $m_x(e)$, we consider four cases. In each case, we use the following two simple facts.

Fact 1: $d_+^d(x) + d_-(x) \leq k$. Note that $d_+^d(x) + d_-(x)$ is exactly the number of edges incident to x that are colored by **OPT**. Thus, Fact 1 follows trivially, since no algorithm can color more than k edges incident to x .

Fact 2: $d_+(x) + d_+(y) \geq k$. This follows from the fact that the edge (x, y) is rejected by the fair algorithm **F**.

Case 1: The parent edge of x belongs to E_- . In this case,

$$\begin{aligned} m_x(e) &\geq \frac{m(x) - C}{d_-(x) - 1} = \frac{d_+(x) - Cd_+^d(x) - C}{d_-(x) - 1} \\ &\geq \frac{d_+(x) - Cd_+^d(x) - C}{k - d_+^d(x) - 1}, \text{ by Fact 1.} \end{aligned}$$

Thus,

$$m_x(e) + m_y(e) \geq \frac{d_+(x) - Cd_+^d(x) - C}{k - d_+^d(x) - 1} + (1 - C)d_+(y).$$

We now make the following calculations, where

- the implication (4) \Rightarrow (3) follows from $d_+^d(x) \leq d_+(x)$.
- the implication (5) \Rightarrow (4) follows from Fact 2.

$$\begin{aligned}
& \frac{d_+(x) - Cd_+^d(x) - C}{k - d_+^d(x) - 1} + (1 - C)d_+(y) \geq C \\
& \Leftrightarrow d_+(x) + (1 - C)d_+(y)(k - d_+^d(x) - 1) \geq Ck \quad (3) \\
& \Leftarrow d_+(x) + (1 - C)d_+(y)(k - d_+(x) - 1) \geq Ck \quad (4) \\
& \Leftarrow d_+(x) + (1 - C)(k - d_+(x))(k - d_+(x) - 1) \geq Ck \quad (5)
\end{aligned}$$

If one allows $d_+(x)$ to be any real number, then we claim that the left hand side of (5) attains its minimum value of Ck when $d_+(x) = k - \sqrt{k}$. To see this, consider the following quadratic polynomial $f(t)$ which corresponds to the left-hand side of (5) with $d_+(x)$ replaced by t :

$$f(t) = (1 - C)t^2 + ((2k - 1)C - (2k - 2))t + (1 - C)(k^2 - k)$$

Note that the coefficient $1 - C$ of t^2 in $f(t)$ is strictly positive. This implies that $f''(t) > 0$ and so f must have a global minimum when $f'(t) = 0$. We claim that $f'(k - \sqrt{k}) = 0$. This follows from the following calculations:

$$\begin{aligned}
f'(k - \sqrt{k}) &= 2(1 - C)(k - \sqrt{k}) + (2k - 1)C - (2k - 2) \\
&= (2\sqrt{k} - 1)C - (2\sqrt{k} - 2) \\
&= (2\sqrt{k} - 1) \frac{2\sqrt{k} - 2}{2\sqrt{k} - 1} - (2\sqrt{k} - 2) = 0.
\end{aligned}$$

We now evaluate $f(k - \sqrt{k})$:

$$\begin{aligned}
f(k - \sqrt{k}) &= (1 - C)(k - \sqrt{k})^2 + ((2k - 1)C - (2k - 2))(k - \sqrt{k}) + (1 - C)(k^2 - k) \\
&= (\sqrt{k} - k)C + 2(k - \sqrt{k}) \\
&= (\sqrt{k} - k) \frac{2\sqrt{k} - 2}{2\sqrt{k} - 1} + 2(k - \sqrt{k}) \\
&= \frac{2k - 2\sqrt{k} - 2k\sqrt{k} + 2k}{2\sqrt{k} - 1} + \frac{2(k - \sqrt{k})(2\sqrt{k} - 1)}{2\sqrt{k} - 1} \\
&= k \frac{2\sqrt{k} - 2}{2\sqrt{k} - 1} = Ck.
\end{aligned}$$

This proves that $f(t) \geq Ck$ for any real number t . In particular, the inequality (5) is satisfied for all values of $d_+(x)$ and hence $m_x(e) + m_y(e) \geq C$.

Case 2: The parent edge of x belongs to E_+^s . In this case,

$$\begin{aligned}
m_x(e) &= \frac{m(x)}{d_-(x)} = \frac{(d_+(x) - 1) - Cd_+^d(x)}{d_-(x)} \\
&\geq \frac{d_+(x) - Cd_+^d(x) - 1}{k - d_+^d(x)}, \text{ by Fact 1.}
\end{aligned}$$

Thus,

$$m_x(e) + m_y(e) \geq \frac{d_+(x) - Cd_+^d(x) - 1}{k - d_+^d(x)} + (1 - C)d_+(y).$$

We now make the following calculations, where

- the implication (7) \Rightarrow (6) follows from the fact that $d_+^d(x) = d_+(x) - d_+^s(x) \leq d_+(x) - 1$ and that the left hand side of (6) is decreasing in $d_+^d(x)$.
- the implication (8) \Rightarrow (7) follows from Fact 2.

$$\frac{d_+(x) - Cd_+^d(x) - 1}{k - d_+^d(x)} + (1 - C)d_+(y) \geq C$$

$$\Leftrightarrow d_+(x) - 1 + (1 - C)d_+(y)(k - d_+^d(x)) \geq Ck \quad (6)$$

$$\Leftrightarrow d_+(x) - 1 + (1 - C)d_+(y)(k - (d_+(x) - 1)) \geq Ck \quad (7)$$

$$\Leftrightarrow d_+(x) - 1 + (1 - C)(k - d_+(x))(k - (d_+(x) - 1)) \geq Ck \quad (8)$$

If we allow $d_+(x)$ to be any real number, then one can show that the left hand side of (8) attains its minimum value of Ck when $d_+(x) = k - \sqrt{k} + 1$. This can be shown by calculations similar to those in Case 1. In particular, the inequality (8) is satisfied for all values of $d_+(x)$ and hence $m_x(e) + m_y(e) \geq C$.

Case 3: The parent edge of x belongs to E_+^d . In this case, we have

$$\begin{aligned} m_x(e) &= \frac{m(x)}{d_-(x)} = \frac{(d_+(x) - 1) - C(d_+^d(x) - 1)}{d_-(x)} \\ &\geq \frac{d_+(x) - Cd_+^d(x) + C - 1}{k - d_+^d(x)}, \text{ by Fact 1.} \end{aligned}$$

Thus,

$$m_x(e) + m_y(e) \geq \frac{d_+(x) - Cd_+^d(x) + C - 1}{k - d_+^d(x)} + (1 - C)d_+(y).$$

Now, we do the following calculations, where

- the implication (10) \Rightarrow (9) follows from $d_+^d(x) \leq d_+(x)$.
- the implication (11) \Rightarrow (10) follows from Fact 2.

$$\frac{d_+(x) - Cd_+^d(x) + C - 1}{k - d_+^d(x)} + (1 - C)d_+(y) \geq C$$

$$\Leftrightarrow d_+(x) + C - 1 + (1 - C)d_+(y)(k - d_+^d(x)) \geq Ck \quad (9)$$

$$\Leftrightarrow d_+(x) + C - 1 + (1 - C)d_+(y)(k - d_+(x)) \geq Ck \quad (10)$$

$$\Leftrightarrow d_+(x) + C - 1 + (1 - C)(k - d_+(x))(k - d_+(x)) \geq Ck \quad (11)$$

We now claim that inequality (11) is satisfied for all values of $d_+(x)$. If $d_+(x) = k$, then the inequality reduces to $k + C - 1 \geq Ck$, which is true for all $k \geq 1$. If $d_+(x) < k$ then the left-hand side of (11) is at least $d_+(x) + (1 - C)(k - d_+(x))(k - d_+(x) - 1)$. As shown in Case 1, this quadratic polynomial is at least Ck for all possible values of $d_+(x)$. Hence, $m_x(e) + m_y(e) \geq C$.

Case 4: The parent edge of x does not exist. In this case,

$$m_x(e) = \frac{m(x)}{d_-(x)} = \frac{d_+(x) - Cd_+^d(x)}{d_-(x)} > \frac{d_+(x) - Cd_+^d(x) + C - 1}{k - d_+^d(x)}.$$

Thus, $m_x(e) + m_y(e) \geq C$ follows from the same calculations as in Case 3.

Assume now that F is a randomized algorithm. The above analysis holds for any coloring that F may produce. Hence, for any coloring produced by F , the number of colored edges is at least $(2\sqrt{k} - 2)/(2\sqrt{k} - 1)$ times the number of edges colored by OPT . Clearly, this means that the expected number of edges colored by F is at least $(2\sqrt{k} - 2)/(2\sqrt{k} - 1)$ times the number of edges colored by OPT . \square

We will show that the lower bound of Theorem 6 is essentially tight by providing a matching upper bound on the competitive ratio of **Next-Fit** when k is a square number. To this end, we will use the following result from [7].

Lemma 1 (Favrholt and Nielsen [7]). *If a graph is colored in such a way that each color is used exactly n or $n + 1$ times for some $n \in \mathbb{N}$, then there exists an ordering of the edges such that **Next-Fit** produces an equivalent coloring.*

The following corollary follows easily from Lemma 1.

Corollary 2. *Consider a graph, $G = (V, E)$, and a coloring, \mathcal{C} , of G using at most k colors. Let H be a graph consisting of k disjoint copies of G . There exists an ordering of the edges of H such that, for each of the k copies of G in H , the coloring produced by **Next-Fit** is equivalent to \mathcal{C} .*

Proof. Let G_1, G_2, \dots, G_k denote the k copies of G . Furthermore, let $\mathcal{C}_1, \mathcal{C}_2, \dots, \mathcal{C}_k$ be the k colorings that can be obtained from \mathcal{C} by cyclic permutations of the colors $1, 2, \dots, k$. If, for $1 \leq i \leq k$, G_i is assigned the coloring \mathcal{C}_i , we obtain a coloring of H where all colors are used the same number of times. The result now follows from Lemma 1. \square

Note that Corollary 2 implies that if \mathcal{G} is some family of graphs and \mathcal{G} is closed under disjoint union, then **Next-Fit** has the worst possible competitive ratio among fair algorithms for $EDGE\text{-}k\text{-}COLORING(\mathcal{G})$. This can be seen in the following way: For any graph, G , and any coloring, \mathcal{C} , of G produced by a fair algorithm, the adversary can make k copies of G and give the edges of the resulting graph in an order such that **Next-Fit** produces a coloring equivalent to \mathcal{C} on each copy of G . Hence, for any sequence, E_G , of edges and any fair algorithm F , there is a sequence, E_H , of edges, such that **Next-Fit** uses just as many colors on E_H as F does on E_G , and the optimal number of colors is the same for both sequences.

Even though **TREE** is not closed under disjoint union, a forest consisting of k trees may be made into a single tree by revealing $k - 1$ edges connecting the k trees. Since this will add at most $k - 1$ to the number of edges colored by **Next-Fit**, we may still apply Corollary 2 for the class **TREE**.

Theorem 7. *If $k = n^2$ for some integer $n \geq 2$, then **Next-Fit** is a worst possible fair algorithm with*

$$C_{\text{NF}}^{\text{TREE}}(k) = \frac{2\sqrt{k} - 2}{2\sqrt{k} - 1}.$$

Proof. The lower bound follows from Theorem 6. For the upper bound, we define a tree $T = (V, E)$ and a subset $E' \in E$. We specify a k -coloring, \mathcal{C} , of E' with the property that each edge in $E \setminus E'$ is adjacent to edges of all k colors.

The tree T consists of N *bunches* of stars, for some large N . Each bunch contains a *large* star with $k - \sqrt{k}$ edges colored with $\mathcal{C}_{1, k - \sqrt{k}}$ and $\sqrt{k} - 1$ *small* stars, each with \sqrt{k} edges colored with $\mathcal{C}_{k - \sqrt{k} + 1, k}$. The center vertex of the large star in bunch i , $1 \leq i \leq N$, is called v_i . This finishes the description of E' and its coloring. For each bunch of stars, $E \setminus E'$ contains an edge between v_i and the center vertex of each of the small stars in the bunch. For $1 \leq i \leq N$, the i th bunch is connected to the $(i + 1)$ th bunch by an edge from v_{i+1} to the center vertex of one of the small stars in the i th bunch. This finishes the description of T . Note that, after assigning the coloring \mathcal{C} to E' , none of the edges in $E \setminus E'$ can be colored.

The adversary will use k disjoint copies, T_1, \dots, T_k , of T . For each T_i , let E'_i denote the set of edges corresponding to E' and let $T'_i = (V, E'_i)$. If the edges of $E'_1 \cup E'_2 \cup \dots \cup E'_k$ are given first, it follows from Corollary 2 that they can be given an order such that the coloring produced by **Next-Fit** on each T'_i is equivalent to \mathcal{C} . Afterwards, no other edges can be colored.

Finally, the k disjoint trees are connected, using $k - 1$ edges between vertices that have degree one in the trees. The resulting tree is called \mathcal{T} . Since $k \geq 4$, we must have $\sqrt{k} + 2 \leq k$ and so the maximum degree of the graph is k . Thus, since the graph has no cycles, **OPT** colors all edges of the graph.

Next-Fit colors $kN(k - \sqrt{k} + (\sqrt{k} - 1)\sqrt{k}) + k - 1 = kN(2k - 2\sqrt{k}) + k - 1$ edges and rejects $k(N(\sqrt{k} - 1) + N - 1) = kN\sqrt{k} - k$ edges. Since **OPT** colors all edges in the graph, $\text{OPT}(\mathcal{T}) = kN(2k - \sqrt{k}) - 1$. Thus,

$$\text{Next-Fit}(\mathcal{T}) = \frac{2\sqrt{k} - 2}{2\sqrt{k} - 1} \text{OPT}(\mathcal{T}) + k - \frac{1}{2\sqrt{k} - 1}.$$

Since N can be arbitrarily large, the result follows. \square

We will briefly consider the case where k is not a square number. Any fair algorithm for **EDGE-1-COLORING(TREE)** is just the greedy matching algorithm. It is observed in several papers that this algorithm is $\frac{1}{2}$ -competitive (for all input graphs) and that no deterministic algorithm can do better, even when the input graph is a tree. If $k \geq 2$, but not a square number, then the lower bound from Theorem 6 can be slightly improved by using the fact that $d_+(x)$ must be an integer. In particular, one can show that all fair algorithms for **EDGE- k -COLORING(TREE)** are at least $\frac{1}{2}$ -competitive for $k = 2$ and $\frac{2}{3}$ -competitive for $k = 3$. Since these bounds match the upper bound from Theorem 3, we conclude that all fair algorithms have the same competitive ratio when $k \leq 3$.

If $k \geq 4$ (but not necessarily a square number), one can obtain the following upper bound by rounding \sqrt{k} appropriately in the proof of Theorem 7: $C_{\text{NF}}^{\text{TREE}}(k) \leq \frac{\frac{k}{\lceil \sqrt{k} \rceil} + \lceil \sqrt{k} \rceil - 2}{\frac{k}{\lceil \sqrt{k} \rceil} + \lceil \sqrt{k} \rceil - 1}$. In particular, for any fixed $k \geq 4$, the competitive ratio of **First-Fit** is better than the competitive ratio of **Next-Fit** for **EDGE- k -COLORING(TREE)**.

5 Open Problems

Finding optimal online algorithms for **EDGE- k -COLORING** in general and on other classes of graphs is an interesting open problem. We believe that the techniques used in the proofs of Theorems 4 and 6 can be generalized to, e.g., graphs of bounded degeneracy. In particular, graphs of bounded degeneracy can be oriented so that each vertex has bounded outdegree and the resulting digraph is acyclic. This makes it possible to use strategies for redistributing the surplus similar to the ones we have used for trees.

References

1. Amotz Bar-Noy, Rajeev Motwani, and Joseph (Seffi) Naor. The greedy algorithm is optimal for on-line edge coloring. *Inf. Process. Lett.*, 44(5):251–253, 1992.
2. Allan Borodin and Ran El-Yaniv. *Online Computation and Competitive Analysis*. Cambridge University Press, 1998.
3. Joan Boyar and Lene M. Favrholdt. The relative worst order ratio for online algorithms. *ACM Trans. Algorithms*, 3(2):22, 2007.
4. Joan Boyar, Lene M. Favrholdt, and Kim S. Larsen. The relative worst order ratio applied to paging. *Journal of Computer and System Sciences*, 73:818–843, 2007.
5. Zhi-Zhong Chen, Sayuri Konno, and Yuki Matsushita. Approximating maximum edge 2-coloring in simple graphs. *Discrete Applied Mathematics*, 158(17):1894–1901, 2010.
6. Martin R. Ehmsen, Lene M. Favrholdt, Jens S. Kohrt, and Rodica Mihal. Comparing first-fit and next-fit for online edge coloring. *Theor. Comput. Sci.*, 411(16–18):1734–1741, 2010.
7. Lene Monrad Favrholdt and Morten Nyhave Nielsen. On-line edge-coloring with a fixed number of colors. *Algorithmica*, 35(2):176–191, 2003.
8. Uriel Feige, Eran Ofek, and Udi Wieder. Approximating maximum edge coloring in multigraphs. In *APPROX, volume 2462 of LNCS*, pages 108–121, 2002.
9. Marcin Kamiński and Łukasz Kowalik. Approximating the maximum 3- and 4-edge-colorable subgraph. *Algorithm Theory-SWAT 2010*, pages 395–407, 2010.
10. Anna R. Karlin, Mark S. Manasse, Larry Rudolph, and Daniel Dominic Sleator. Competitive snoopy caching. *Algorithmica*, 3:77–119, 1988.
11. Hal A Kierstead. Coloring graphs on-line. In *Online Algorithms*, pages 281–305. Springer, 1998.
12. Adrian Kosowski. Approximating the maximum 2- and 3-edge-colorable subgraph problems. *Discrete Applied Mathematics*, 157(17):3593 – 3600, 2009.
13. Romeo Rizzi. Approximating the maximum 3-edge-colorable subgraph problem. *Discrete Mathematics*, 309(12):4166 – 4170, 2009.

14. Daniel D. Sleator and Robert E. Tarjan. Amortized efficiency of list update and paging rules. *Commun. ACM*, 28(2):202–208, 1985.
15. Andrew Chi-Chih Yao. Probabilistic computations: Toward a unified measure of complexity (extended abstract). In *FOCS*, pages 222–227, 1977.